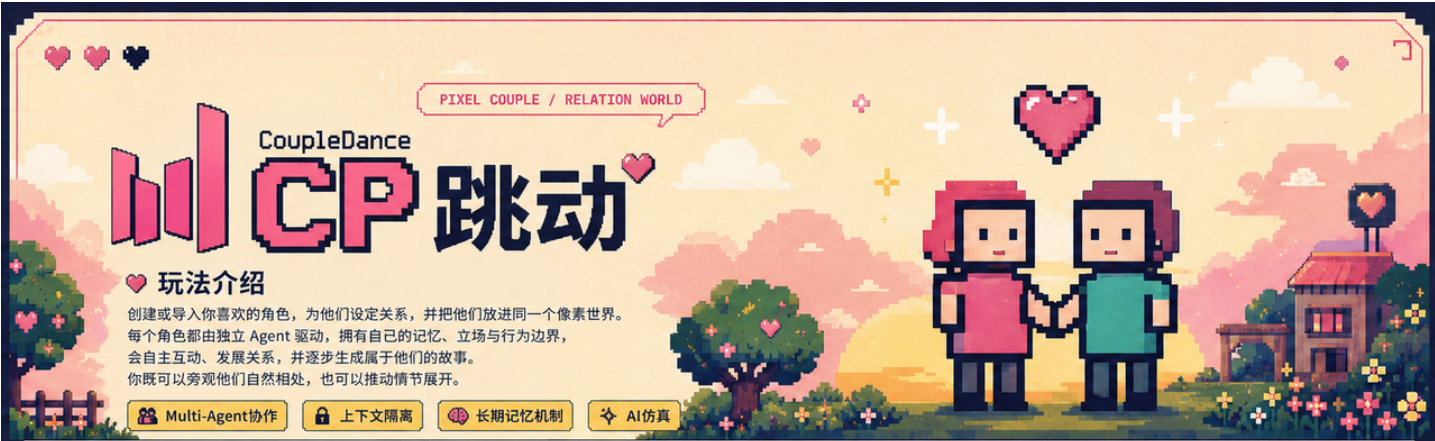


CP 跳动 · Couple DANCE 产品介绍手册 公开版本



把你最喜欢的角色放在一起，看 TA 们亲自演出你想嗑的故事。

github: <https://github.com/qiukuizhenpiaoliang-dot/cp-dance>

1. 读懂产品

1.1 产品一句话

💖 CP 跳动 · Couple DANCE 是一款面向角色厨、CP 爱好者和同人爱好者的 AI 像素角色扮演游戏。

在这里，你喜爱的角色们会自由地交谈、拌嘴或者贴贴，或许漫步在你的桌面上，或许去开启一场全新的冒险。

你可以创建或导入自己喜欢的角色，为他们选择关系，把原作 CP、自设角色或不同作品中的角色放进同一个世界，观察他们如何相识、相处，并发展出属于自己的故事。

每个角色都由独立的 Agent 驱动，拥有自己的记忆、立场和行为边界。Agent 会根据当下发生的事情，去模拟角色决定说什么、做什么动作和表情，以及是否愿意靠近另一个角色。

在自然世界中，你可以把角色作为桌宠，看角色自由聊天、互动、贴贴、闹别扭，或者各自做自己的事

在导演剧场中，你可以输入同人设定、故事背景和剧情 Prompt，安排重逢、误会、相爱相杀、一见钟情等事件，也可以选择 HE、BE、TRUE END 等结局倾向。

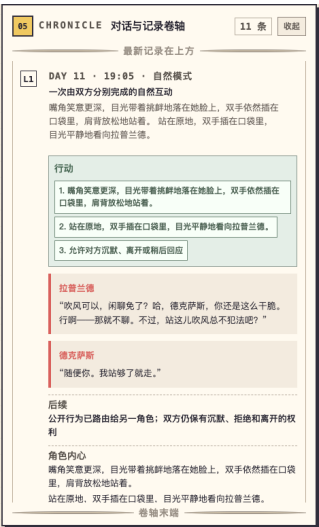
导演 Agent 会负责构造舞台、生成背景图片和外部事件，但角色最终如何应对，依然由他们自己决定。



角色的状态，不仅反映在动作上，通过角色状态卡片也会告诉你 TA 在偷偷想什么



所有角色的对话、动作都会被记录到卷轴上，可以随时翻阅。



在结束一个世界后，支持查看角色互动轨迹、以及可以一键生成角色故事文章和结局定格shot，并且可以将页面一键转发分享。

角色的关系和记忆也可以动态观察到。



1.2 为什么要做 CP 跳动

1.2.1 同人创作的核心

做同人创作最让人上头的，不是敲定好的结局。是指尖将碰未碰时的停顿，是嘴硬到耳根发红的逞强，是误会里错开的眼神，是往前半步又收回的犹豫，是那些只属于「这两个人」的瞬间💖。观众想看的从来不是套着人名的通用模板，是他们真的会这样拌嘴，这样别扭地靠近，这样阴差阳错地错过的角色张力。

传统同人游戏把所有走向锁死在预设分支里，普通 AI 聊天又只剩你和单角色的对话框，明明是磕 CP，却只能单向对话。

而「CP 跳动」能够做到：角色不再只对着玩家单向回应，你磕的 CP 会进入同一个世界，感知对方的存在。没有固定剧本，没有有限分支。每一段试探、每一次别扭、每一场心动，都只属于这两个角色。TA 们会自己走出独一份的故事，而你能亲眼见证每一秒的发生。

「CP 跳动」也支持单人世界，最大人数限制为3人，未来可拓展人数。

1.2.2 痛点与产品答案

现有 AI 角色扮演和同人创作的痛点	CP 跳动给出的答案：
创作成本高：没有灵感，不知道如何入手，不会画角色图片或者写不出来想看的故事	创作成本低：玩家可以直接输入或者一键导入角色资料，角色自动进入故事。结束后可以一键导出记录生成同人文。

- **角色难以贴近设定：** 设定写得很长，但进入长对话后仍会漂移，称呼、边界、冲突方式和对另一人的独特看法难以稳定。
 - **互动缺少观测性：** 无法做到可视化，看不到角色如何交互。
 - **无法长时间运行：** 全量历史不断塞回模型既昂贵又容易把旧噪声当成当前事实；如果直接摘要，又可能把关键承诺、未回答问题或人物边界压没。这些问题，是导致角色脱离设定（OOC）的原罪。
 - **游戏体验单一：** 只有文字或者图片的单一体验，可玩性和拓展性差
 -
2. **引入角色桌宠系统：** 内置角色生成 agent，上传一张角色图片就可以生成能交互的桌宠，持续生成丰富的表情和动作，可以全职在桌面上研究嗑 CP 了

 3. **角色本身是自由的：** 每个 Character Agent 只控制自己，拥有沉默、拒绝、反悔自由行动的权利。
 4. **角色稳定性机制：** 支持输入角色设定，三层记忆存储、索引检索与上下文压缩从架构层共同保证连续性，让你的 cp 避免 OOC！
 5. **给用户设定故事背景的能力：** 玩家一句话说出背景和想看的故事，AI 模拟角色进行自然地生活。
 6. **体验闭环且丰富：** 从「角色创建→考据蒸馏→动作生成→关系绑定→双模式互动→桌宠形态→存档重访→生成同人图文」形成了完整的体验闭环。

1.2.3 产品为谁而做

用户	用户真正想要什么	CP 跳动提供的体验
同人女 / CP 爱好者	不只看一句“他们在一起了”，而是持续观察相处细节	看角色自己聊天、靠近、闹别扭、和好，让关系在一次次互动中慢慢发展，最后提供角色的回忆地图和结局图，供细细品味。
角色厨 / 原作党	角色说话做事要贴近原作，不被通用人格覆盖	可以导入和补充原作资料，亲自修改角色档案，也支持 skill 联网蒸馏角色，让角色尽量保持原本的性格、经历和说话方式
自设 / OC 创作者	让自己的角色离开设定表，进入可观察的生活	把 OC 做成能行动、会说话、记得经历的像素角色，既能独自生活，也能和其他角色建立关系
同人作者 / 跑团与剧情创作者	快速验证宏大设定中角色会如何反应	输入世界观、剧情大纲或突发事件，让角色自行演绎；还可以设置 HE、BE 或 TRUE END 等故事倾向。结束后可以通过回忆地图复盘重大情节
桌宠玩家	希望角色真正“住”在桌面上，而不是只播放动画	让角色常驻桌面，会回应点击和拖拽，也会根据彼此的距离自主说话、靠近、沉默或离开

2. 产品体验

2.1 体验流程

CP 跳动把一次体验拆成“建立角色—绑定关系—选择世界—持续运行—保存与重访”五个连续阶段。

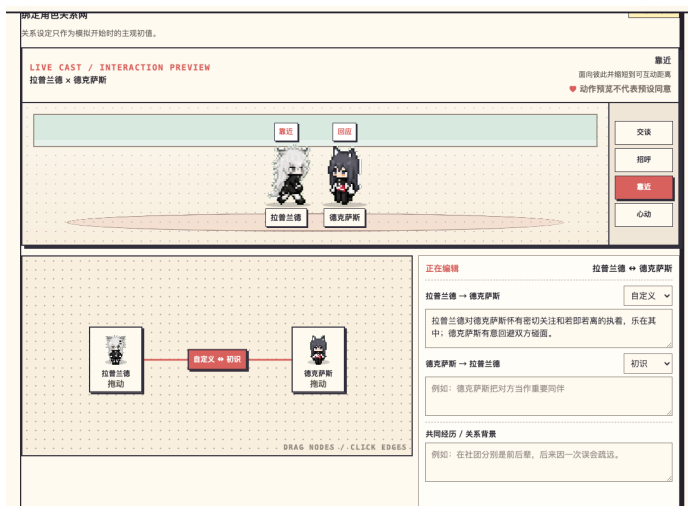
1. **从入口开始。** 玩家可以读取已有世界、从已有角色档案组队，或通过 START UP 从零创建。



2. 建立角色并让角色动起来。填写身份、背景、性格与角色扮演资料；可选择角色考据 Skill，Skill 可以搜索网络内容帮你蒸馏角色；上传参考图并生成可交互动作，除了基础设定的动作，还支持自定义添加其他动作、表情。



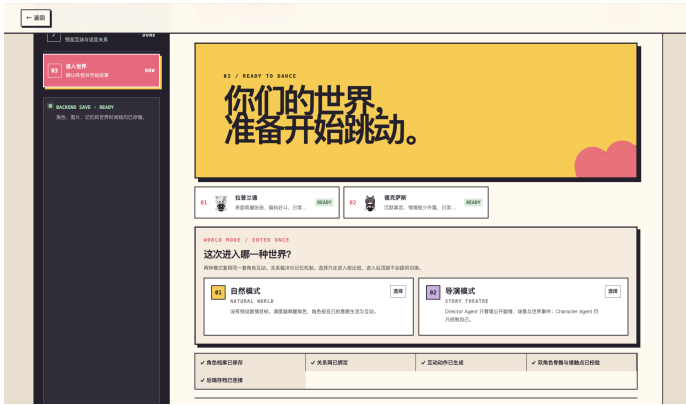
3. 绑定关系。为每一组角色（1-3 人）分别填写关系与共同经历，并可以预览已生成的动作。



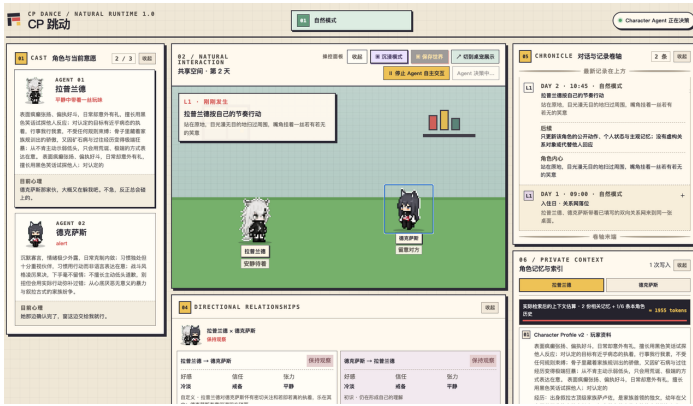
4. 选择世界模式。选择自然模式或者导演模式。

在自然世界中，看角色自由聊天、互动、贴贴、闹别扭，或者各自做自己的事。

在导演剧场中，你可以输入同人设定、故事背景和剧情 Prompt，安排各种剧情、选择 HE、BE、TRUE END 等结局倾向。



5. 开始世界的运行。玩家可以沉浸地在剧场网页里看角色互动，也能让角色作为桌宠在桌面上生活。停止并保存后，世界记忆、关系、公开事件、空间、动作索引和剧情摘要写入存储；下次读取时继续这个世界，而不是重新开一段无记忆聊天。



5. 结束世界和回忆世界。停止并保存后，可以生成回忆录。

回忆录包括角色关键节点的回忆地图，支持通过卷轴记录的内容一键生成完整的故事文章，以及结局的定格插图。

所有内容支持一键打包下载和分享。



2.2 玩法一：制作桌宠，让角色真正动起来

cp 跳动可以把角色档案、视觉参考、动作资产、空间骨骼与 角色扮演 Agent 身份组合成一个可持续运行的角色实体。

2.2.1 人物设定资料

玩家可以纯手工填写角色档案，如果不想花费时间，也可以使用内置的角色考据 Skill。Skill 会搜索网络内容，由玩家确认正确来源，再读取可见正文并整理为带来源、置信度和限制的草稿，最终蒸馏为角色档案。

角色档案不只是在创建页展示的资料，更是每次 Character Agent 决策时的稳定身份层，是确保你角色不跑偏的关键文件。

角色名字

丰川祥子

性格设定 (可选)

清冷孤傲的极致完美主义者，对音乐与团队有着近乎偏执的严苛要求。说话尖锐直接，从不留情面。习惯性用冰冷的外壳包裹自己，自尊心极强，绝不轻易示弱低头，看似冷漠不近人情，实则内心敏感脆弱，极度惧怕孤独与否定，不擅长坦诚表露真

背景介绍

经历：出身优渥家庭，自幼接受精英式钢琴教育，音乐功底扎实深厚。高中时与高松灯、千早爱音、要乐奈、椎名立希组建乐队 CRYCHIC，担任队长与键盘手，将全部心血倾注于乐队，坚信能一同走下去，却在普通正式 Live 前夕因分歧与内部矛盾，被逐出乐队。

角色扮演资料 (可选)

说话口吻：语气清冷疏离，语速平稳，说话直接尖锐，常一针见血戳中要害，极少流露多余情绪；谈及音乐与乐队规则时会变得格外严格，情绪激动时语速会上极难察觉的翻调；面对认可的人语气会稍有松动，但依旧保持克制与距离，喜欢末尾带deseawa的敬语。
称呼：对他人多直呼全名或队内称谓，几乎不用亲昵称呼；对旧友也保持着疏离的直呼全名，对陌生人心敏
习惯动作：思考时会用指尖无意识轻敲桌面或琴键；情绪压抑时会紧蹙眉头边缘；指尖发白也不吭声；独处时会无意识摩挲旧乐队徽章；演奏前会反复调整琴凳与音量高

应用角色考据 Skill

所属作品 / 时期 (可选)

mygo

提交人物资料

找到 2 个经过人物筛选的候选，游戏、歌曲、游戏文和作品页已链接。

选择正确人物

☒ 维基百科 - 编辑角色

☐ 维基百科 - 编辑角色

丰川祥子

编辑百科分类：2月14日，BanG Dream!角色、不辛、丝等、中二病、丰川...

编辑百科页面 · Zh

最多选择 3 个来源：只有你选择的页面会被读取。

混合来源已选来源 (1)

丰川祥子

丰川祥子 (日语：豊川 / とがわ) 祥子 (さきこ) 是由日本娱乐公司(Bushiroad)策划的次世代少女乐队企划 (BanG Dream!)

点击“应用已确认内容”后，确认项才进入角色档案。

01 / 编辑并确认资料

☒ 身份

确认

丰川祥子是Ave Mujica的键盘手，代号Oblivions。

编辑百科 · 原文证据：“Ave Mujica的键盘手，代号Oblivions。”

☐ 背景

确认

祥子为月之森女子学园初三学生，后转学至羽丘女子学园。

编辑百科

☐ 行为锚点

确认

祥子退出CRYCHIC后主导组建Ave Mujica，并负责剧本和作曲。

编辑百科 · 原文证据：“退出CRYCHIC后，她主导组建了Ave Mujica，担任键盘手、负责剧本和作曲。”

☐ 价值观

确认

祥子重视乐队纪律，要求成员隐藏真实身份。

编辑百科 · 原文证据：“对乐队成员提出了诸如禁止私人摄影与不得暴露真实身份等严格的要求。”

☒ 关系

确认

祥子与若叶睦是青梅竹马，关系复杂。

编辑百科

☐ 行为锚点

确认

在手游剧情中，当被逐出CRYCHIC或时，祥子会恨过花。

编辑百科

☐ 表达方式

确认

祥子以乐队主导者身份对初华说话，强调契约和责任感，如“即使再怎么敷衍，我也不再会再让你逃跑了。你将人生托付给了我们，我们不是签下契约了吗？”

编辑百科

☐ 行为锚点

确认

在MyGO和Mujica动画中多次哭泣，包括因春日影歌唱、母亲离世、CRYCHIC解散、与父亲争吵、被墨斯路辱骂等场景。

编辑百科 · 原文证据：“在MyGO与Mujica动画中有大量春日影歌唱的哭泣镜头，包括MyGO动画中第三章春日影为春日影作作的哭泣的哭泣，第七章再次对春日影时春日影的哭泣；Mujica动画中第一章春日影世界的哭泣，CRYCHIC解散时的痛哭大哭，与父亲吵架后在前往武道馆的车上抹眼泪，第五章春日影“小祥，你幸福吗”时的落泪，第七章被墨斯路辱骂后在深夜在深夜门外时的哭泣、Cry”

☐ 表达方式

确认

祥子以乐队主导者身份对初华说话，强调契约和责任感，如“即使再怎么敷衍，我也不再会再让你逃跑了。你将人生托付给了我们，我们不是签下契约了吗？”

编辑百科

方向性关系证据

☒ 丰川祥子 →

确认

若叶睦

编辑百科

☒ 丰川祥子 →

确认

长崎素世

编辑百科

☒ 丰川祥子 →

确认

三角初华

编辑百科

☒ 丰川祥子 →

确认

高松灯

编辑百科

2.2.2 上传参考图，生成可交互角色

玩家上传参考图（推荐像素 q 版图）后，角色生成 agent 通过图像编辑通道（经过多轮评测，推荐使用 gpt-image2 模型）三朝向基础动作表，覆盖待机、行走、招呼、哭泣与心动等。

系统从动作关键帧中提取头、胸、髋、左右手、左右脚七个骨骼点，为靠近、摸头、牵手、拥抱、贴贴和共同移动提供可验证的空间基础，角色 Agent 可以真实地操纵角色的身体进行交互。

基础动作之外，角色还可以获得害羞、生气、说话、倾听等增量动作。



页面支持下载已经制作好的交互图，并提供预览页面。

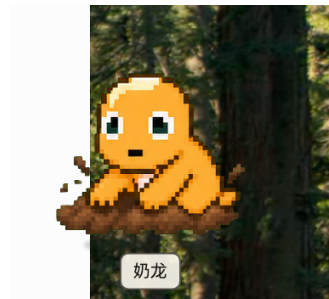


2.2.3 开启桌宠

进入自然世界后，点击桌宠模式，就可以在电脑上一边工作一边惬意看 CP 谈恋爱💕

玩家可以轻点或拖动角色。轻点只产生短暂触发，角色可以回应、沉默或继续原动作；拖拽落点会成为公开空间事实，受影响角色分别判断是否吃惊、害羞、脸红、生气、靠近或离开。拖近不等于同意接触，拖远也不会直接扣好感。桌面气泡显示角色扮演 Agent 产生的台词。





2.3 玩法二：自然模式

😊 自然模式没有预设剧情目标，因此，玩家可以只是把 CP 捏好，打开世界，看他们贴贴、吵架、互相试探、忽然聊起旧事，或者各自做自己的事。单角色世界里，角色会独自探索、观察、休息并积累自己的记忆。

当角色提出牵手、轻触、拥抱、贴贴、摸头、靠肩、共舞或共同移动时，系统不会把发起者的一句话直接变成既成事实。接收角色会获得独立任务，选择接受、犹豫、拒绝或提出替代方案。

所有角色的对话和动作可以在卷轴面板中看到记录，角色的状态在角色面板中可以观察，面板均可折叠，也可以选择沉浸模式大屏观看互动。



2.4 玩法三：导演模式



当玩家想看一场轰轰烈烈的同人，可以进入导演模式：

把 CP 丢进宏大末世、轻松日常、原著背景或跨作品世界。玩家填写核心设定、故事舞台、整体气质、额外限制等，并可以选择结局是自然收束、HE、BE（好坏结局）或寻找 TRUE END。

Director Agent 会建立故事标题、概要与 Plot Beats，制造天气、时间、地点、敌袭、线索、危险、任务、秘密暴露和公开状态变化。它可以控制“世界发生什么”，但不能代替角色说话、规定感情结果、读取私有记忆或绕过拒绝。

玩家运行中可以提交“突然停电”“下一幕去家里”“让剧情更偏向寻找线索”等指令。系统会区分强制世界事实、剧情指导和场景请求，但这些输入依然不能直接写角色的动作或台词。

当场景发生切换，背景 agent 会查询全局背景总索引，若无已有符合场景的背景图，则调用生图 api 生成符合当前故事描述的背景图片传回剧场。

即使玩家选择 HE，系统也不会命令角色原谅、告白或接受接触。结局倾向只影响机会、节奏和外部条件；结局中的角色选择仍然属于角色。

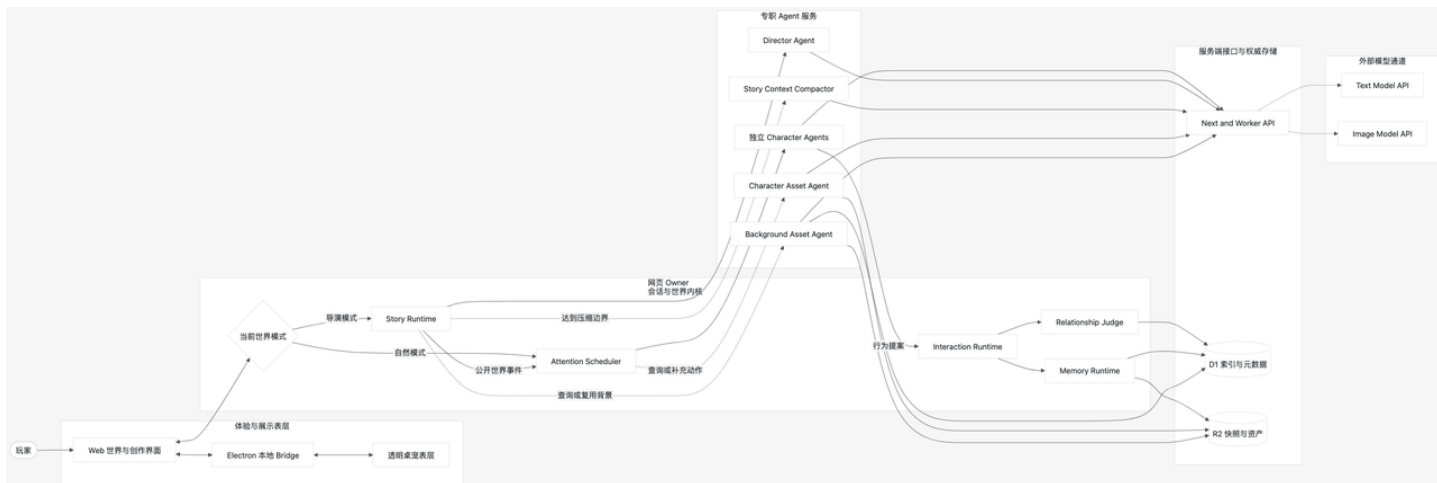


3. 产品是如何实现的：技术设计

3.1 整体架构



项目采用分层多 Agent 架构，以大语言模型和图像生成模型提供 AI 能力，使用 Next.js 与 Electron 构建 Web 和桌宠前端，并通过 Cloudflare D1 与 R2 分别存储结构化数据和角色、背景等资产。



3.2 Multi-Agent 架构设计

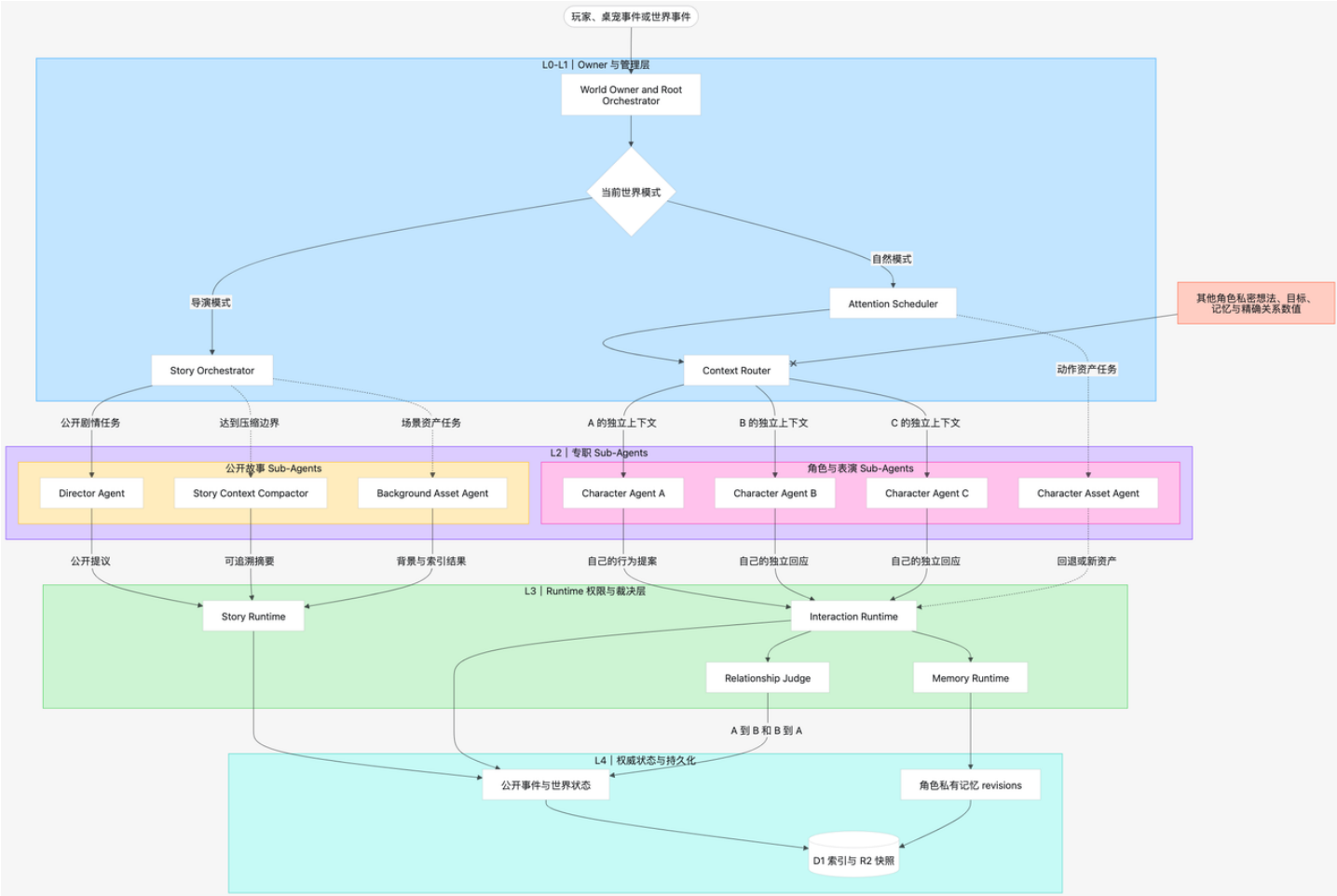


CP 跳动采用 Sub-Agents 架构来管理众多 agent，目的是解决角色串台、导演操控角色、长期故事上下文互相污染的问题。

Sub-agents 架构的价值不只是并发，还有**上下文隔离**。

本次采用一个网页 Owner 会话管理世界，多组专职 Sub-Agent 提出结果，再由本地 Runtime 统一裁决和提交，每个模块只拿到完成自己任务所需的最小上下文，并通过明确协议交接结果，可以有效避免上下文污染。

如：A 的任务只包含 A 的 Profile、A→B 的 Relationship Lens、A 自己的相关记忆，以及 A 能看见的公开事件；B 的 privateThought、目标和私人记忆不会出现。导演的隐藏大纲和结局目标同样不会进入 Character Agent。



模块	负责什么	分层关系
Attention Scheduler / Orchestrator	选择本轮唤醒谁、组装其可读上下文、路由公开回应	Root Orchestrator ：读取世界状态、判断模式并分配领域任务。 Story / Attention 管理层 ：决定调用哪些 Sub-Agent、给谁组装上下文。 专职 Sub-Agents ：分别负责导演、压缩、背景、各角色表演和动作资产。 Runtime 层 ：验证和执行提案，防止 Agent 越权。 状态层 ：提交公开世界、方向关系、私有记忆与后端存档。
Character Agent（角色扮演 agent）	只决定自己的动作、台词、沉默、请求、接受、拒绝与离开	
Character Research Skill	搜索、提炼并生成可编辑资料草稿	
Director Agent	规划 Plot Beats、场景和公开世界事件	
Story Context Compactor	压缩已发生的公开剧情	
Story Runtime	校验导演输出并提交合法世界事实	
Interaction Runtime	校验请求、同意、空间、动作和边界并执行	
Relationship Judge	根据可见结果分别更新 A→B 与 B→A	
Memory Runtime	检索私有记忆、验证 revision、提交安全写入	
Character Asset Agent	补全缺失的单人动作资产	

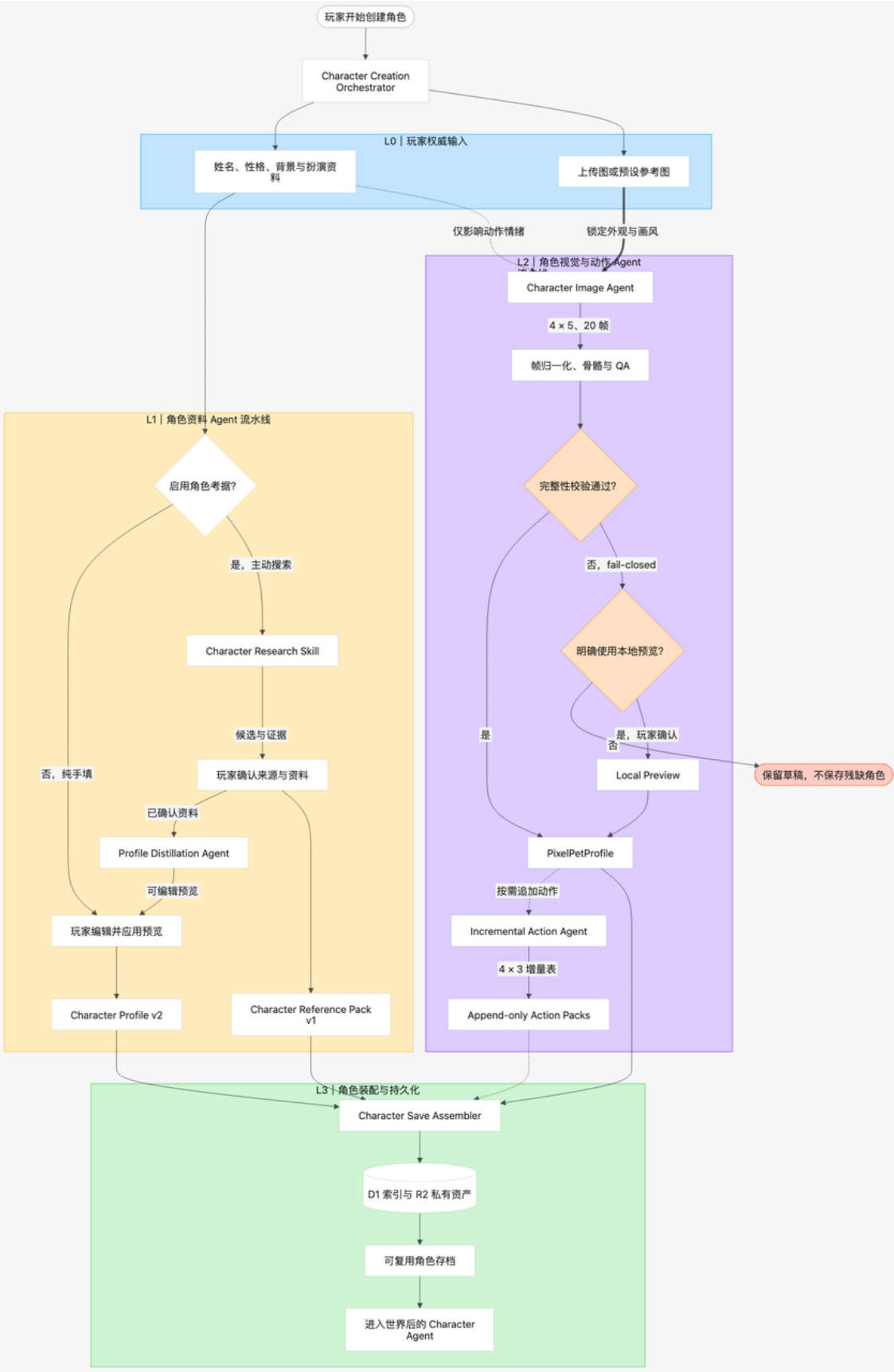
Background Asset Agent	查总索引、复用背景、维护世界索引；显式许可后生成
Desktop Bridge / Surface	把自然世界投影到透明桌面层并回传受限输入

3.3 角色生成模块设计



角色模块负责把一个“名字和想法”，逐步变成一个拥有设定、形象、动作、记忆，并且能够独立互动的角色。

并且在剧情发展到需要新的角色动作、表情时，可以被调用进行生产。



模块	负责什么	什么时候使用	为什么这样设计
角色档案模块	保存姓名、性格、经历、说话方式、行为边界等核心设定	创建原创角色，或者已经掌握完整设定时	玩家填写的资料是角色的最高权威，避免 AI 擅自修改人物
角色搜索 Skill	搜索原作角色资料，整理背景、性格、关系和语言特点	想创建原作角色，但不想手动整理大量设定时，可选择开启	降低创建门槛，解决“查资料、抄设定很无聊”的问题
Profile 提炼 Agent	将零散资料整理成可供角色扮演使用的结构化档案	搜索完成或玩家输入大量资料之后	搜索结果通常很杂，需要提炼成 Agent 真正能理解和使用的角色规则
视觉生成 Agent	根据参考图制作统一规格的像素角色和基础动作	角色档案确认、参考图准备完成之后	将“角色是谁”和“角色长什么样”分开处理，方便单独修改设定或外观 提示词经过3次迭代确定，保证生成保持高质量、完整性和生成一致性
Sprite QA 模块	检查透明背景、帧数、人物位置、动作切分等问题	每次像素角色或动作素材生成之后自动执行	防止坏图、空帧和切分错误直接进入游戏
Rig 与动作模块	识别角色关键身体节点，绑定走路、说话、转身等动作	像素角色通过检查之后	让角色不只是一张立绘，而是可以移动、朝向、靠近和做出真实动作
动作扩展 Skill	为已有角色增加害羞、生气、跳舞等个人动作	角色创建完成后，发现现有动作无法表达新行为时	通过追加动作包扩展能力，不覆盖已经稳定的角色素材
Character Agent	根据角色档案、当前场景、关系和记忆决定说什么、做什么	角色进入自然世界或导演剧场之后	让每个角色独立思考，而不是由一个总模型替所有角色表演
角色记忆模块	保存角色经历、观察、推断和对其他角色的认识	角色开始长期互动后自动运行	保证角色记得发生过的事情，并在长时间运行后保持人格和关系连续性

3.4 角色稳定性设计：记忆系统与信息分层

 在长时间模拟时，产生上下文多且乱，很难保证角色不会偏离设定（OOC，out of character），CP 跳动用三层信息层和压缩上下文机制共同维持角色长期一致性。

3.4.1 三层信息层

- **稳定身份层。**在设定角色时，写入的资料 固定“这个角色是谁”，不会因为一次临时情绪被覆盖。
- **当前场景层。**Public Dialogue、Group Scene、Stage 与最近自身历史保存“刚刚发生了什么、谁在等回答、这回合为什么被唤醒”。
- **长期主观记忆层。**每个角色拥有只属于自己的 `general.txt`、`characters/<agent>.txt` 与 `topics/<topic>.txt`，记录承诺、具体措辞、偏好、边界、共同细节和未完成问题。

3.4.2 约束模型的产出与写入

Character Agent 只有记忆提案权。Memory Runtime 会检查文档归属、最新已读 revision、`baseRevisionId`、可见证据和内容权限。只有满足先读后写且证据可追溯的提案才能形成新 revision；格式错误或越权时，系统只把本回合已经裁决的公开结果作为安全观察写入。

每条记忆还区分 observed、inferred 与 rumor，避免角色把猜测当成事实。另一名角色的私人想法、个人目标、私有记忆和精确关系数据永远不会被送进当前角色上下文。

3.4.3 检索能力设计

每回合，系统按当前对象、显著性、新近度和任务类型确定性排序，最多选择少量最相关记忆，并附带当前角色最近的自身历史。这样可以把有限上下文留给真正影响本次表演的信息，降低 token 成本与延迟，也减少旧噪声造成的人格漂移。

人物一致性来自稳定 Profile、方向性 Lens、证据化 revision 和可审计检索；压缩只用于导演模式已经公开、已经发生的长剧情。

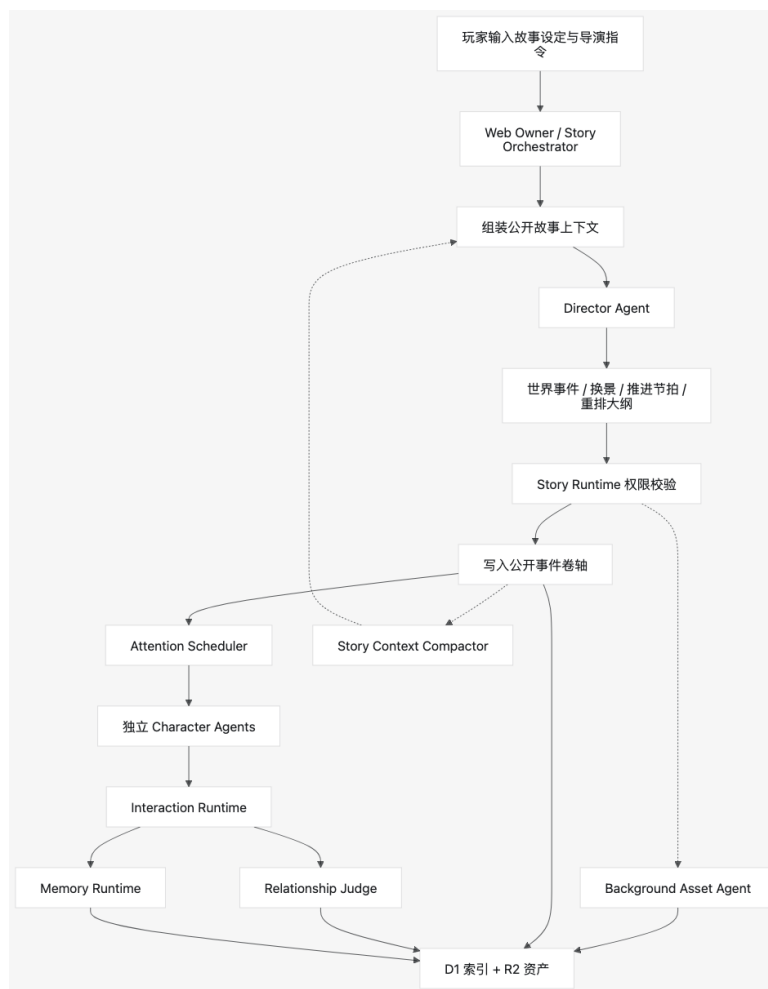
3.4.4 长剧情压缩机制

导演模式中的公开剧情会持续增长，如果不做上下文压缩，故事运行一段时间后，系统只能面对两个结果：

1. **继续把全部历史发送给模型：**上下文会越来越大，导致响应变慢、调用成本上升，最终超过模型的上下文上限，Director Agent 无法继续工作。
2. **直接删除较早的经历：**模型会忘记早期设定、承诺、伏笔和角色边界，出现前后矛盾，例如已经公开的秘密再次被当成秘密、已经解决的冲突重新发生、角色忘记之前的约定。

即使还没有超过模型上限，过长上下文也会产生注意力稀释：重要信息被大量普通对话淹没，Director 更难判断当前剧情重点，容易重复事件、错误推进节拍或让人物性格发生漂移。

所以引入了**长剧情压缩机制**，当故事运行过长时，Story Context Compactor 会压缩已经发生的公开剧情，保留关键事件、未解线索与玩家指令，让 Director 在长期运行后仍能维持故事连续性。



当未压缩尾部达到约 24k tokens / 96 KB 时，系统在安全边界请求软压缩；达到约 40k tokens / 160 KB 时，暂时阻断 Director 继续编排，直到摘要通过本地校验。场景切换、Plot Beat 完成、大纲重排与旧存档恢复也是强制压缩边界。Director 总预算为 64k tokens，压缩目标为 10k-14k tokens，并保留最近最多约 12 个公开节拍与约 6k tokens 的原始尾部，以提升长剧情运行的稳定性。

3.5 一次完整互动是怎样实现的

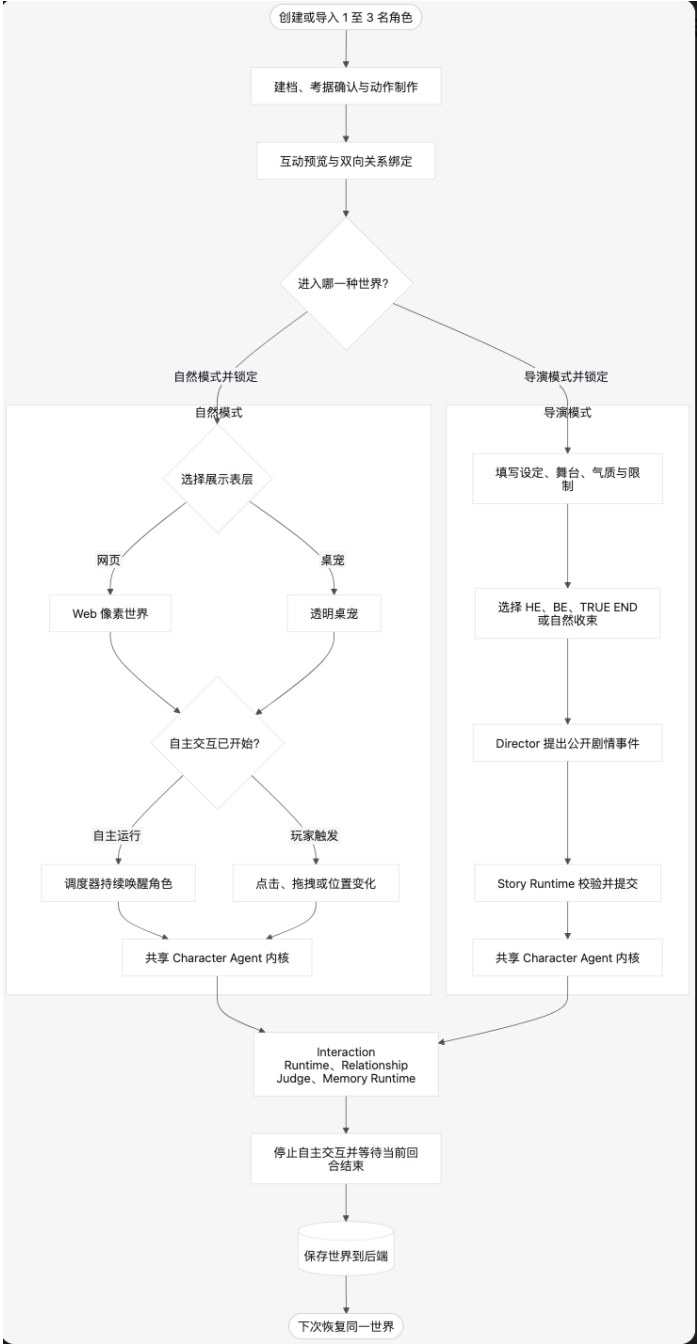
🧐 为了保证角色能够最大程度上还原设定，一次自然互动不是一个 Prompt 直接生成一段故事，而是设计了以下链路：



- 调度。** Attention Scheduler 依据待回答问题、当前距离、参与状态、最近行动与未完成事项选择焦点角色。
- 组装角色上下文。** 加入 Profile、Reference Pack、方向性 Lens、Stage、相关私有记忆、逐字公开对话与 Group Scene。

- 3. **角色决策。** Character Agent 返回自己的一个动作、公开台词、目标对象、是否需要回应，以及受限的记忆提案。
- 4. **独立回应。** 若动作指向他人，系统为每名合法接收者分别建立 Agent 任务；接触与敏感互动只允许单一对象。
- 5. **资产查询。** 查找双方 ready 动作；缺失时安排异步生成，当前回合使用安全回退。
- 6. **空间与同意执行。** Interaction Runtime 验证请求、回应、拒绝锁、距离、路径、骨骼、边界和第三人碰撞。
- 7. **关系结算。** Relationship Judge 只根据已发生的可见结果分别更新两个方向。
- 8. **记忆提交。** Memory Runtime 校验证据与 revision，为每个角色分别写入自己的主观记忆。
- 9. **持久化。** 公开事件、空间、关系、动作索引、记忆 revision 与世界快照在保存时进入后端。

导演模式只在这条链路之前增加“玩家设定 → Story Orchestrator → Director Agent → Story Runtime → 公开世界事实”；角色之后仍走完全相同的自主决策、同意、关系与记忆内核。



3.6 资产索引设计



当生产的资产过多，如何方便 agent 用最快时间找到？

这时候就需要索引登场！

仅仅把图片放进文件夹，Agent 仍不知道它适合“雨夜旧车站”还是“晴天海边栈道”。索引为资产添加地点、时间、天气、氛围、视觉关键词、来源、许可和状态，使 Agent 可以检索、复用和审计。每次新的生成结果必须先写存储、再更新总索引和世界索引。

角色动作索引	
--------	--

	动作资产采用 append-only Action Pack。系统先找精确动作，再找语义接近动作，仍缺失时创建任务，并为当前回合准备基础回退。通过归一化与 QA 的新动作才会登记为 ready；历史资产不被覆盖，角色下一次被唤醒时重新判断是否仍需要使用。
背景总索引与世界索引	背景采用两级索引：全局 <code>background-catalog</code> 记录可以复用的 ready 资产；每个世界的 <code>background-world-index</code> 记录这个世界实际使用过的背景、当前背景与场景绑定。前者解决“有哪些资产”，后者解决“这个世界用过什么、当前该恢复什么”。

4. 思考总结

4.1 想说的话

感谢观看，当前完成版已经完成从角色建档、角色考据、动作角色制作、方向性关系、自然互动、多人共享场景、导演故事、背景复用、桌宠等所有能力。

☒ 从得知这次 AI 产品经理早鸟活动，再到想到创意，到边在腾讯实习边忙中偷闲做出来这个项目，用时 5 天。

做的时候遇到了许多问题，也查了资料，在做的过程中产生了很多自己的思考，对我来说，能把一个原本只存在于脑海里的想法真正从 0 做到 1，已经是一件非常幸福的事情。

现在，我已经连续让自己的 CP 在桌面上自主谈恋爱 36 个小时了！看着他们自己聊天、互动、靠近或者闹别扭，会让我觉得之前投入的时间都变得非常值得。

项目已经开源在 GitHub<https://github.com/qiukuizhenpiaoliang-dot/cp-dance>。

下载后需要在环境里配置两个模型的 api，经过多轮评测，文字模型推荐 deepseekv4，图片模型推荐 image2，具体详情参考 readme 文件。



我是一个对字节产品很感兴趣的人，也一直很希望未来能够进入字节。这个项目的名字和 Logo，也是我在制作过程中埋下的一点小巧思之一～

我也在想，这个项目是否可以对字节未来的一些产品方向提供一点参考价值。

比如，可以应用到：

圈层 UGC 价值：同人圈层天然具备强传播性，用户生成的互动名场面可以一键导出短视频，适配抖音内容生态，形成“创作→传播→引流”的闭环；

IP 衍生价值：可对接番茄小说、动漫 IP，做官方角色的番外互动，成为 IP 内容的新载体；

AI技术价值：抖音、飞书未来是否可以支持用户捏出自己的 AI 分身，让不同分身之间进行自主交互；而分身之间的公开经历、长期记忆、独立判断和关系变化，也许这个项目中 Multi-Agent 协作、上下文隔离和长期记忆也可以被用上？

虽然这还只是一个很初步的想法，但我觉得，AI 的交互未来也许还有很多值得探索的空间.....

4.2 灵感来源与技术借鉴：

想解决的问题	参考的项目	如何借鉴
CP游戏怎么设计？ 只观察、不操控的 CP 互动形态，是否真的有稳定受众？ 非剧情向的日常互动，为什么能让 CP 爱好者持续上头？	任天堂《Tomodachi Life 朋友收集梦想生活》（人森）	最开始萌生做这个项目的想法，其实是刷到很多人用《人森》捏自己喜欢的 CP，游戏里随机触发的几句对话、一个贴贴动作就能让人回味无穷。 虽然游戏本质是预设规则的固定系统，角色人设、互动内容、剧情走向都受限于游戏有限的模板控制，但还是如此让玩家和CP爱好者着迷🐱 但我也明显感觉到，预设系统里的角色终究是模板化的，永远没法真正演出“那两个人”独有的张力。 这让我开始思考：如果用 AI 让每个角色都拥有独立的记忆、性格和行为边界，让他们真的像原作里那样说话、相处、别扭地靠近，会不会就是同人爱好者真正想要的东西？从这个疑问出发，才有了后来的 CP 跳动。
多个AI 角色如何互动？	Stanford 与 Google 的论文 Generative Agents: Interactive Simulacra of Human Behavior 论文提出，生成式角色不能只读取一段人格 Prompt，而需要保存经历、动态检索相关记忆、形成更高层反思，并据此规划后续行为。	CP 跳动借鉴了其中的： • 角色拥有自己的经历记录，而不是只使用全局聊天记录； • 当前行为需要结合人格、最近事件和相关长期记忆； • 记忆不能全部塞进每次请求，需要按照当前任务动态选择； • 长期经历需要被整理成更稳定的高层信息。 但 CP 跳动没有直接复制“观察—反思—计划”流水线。恋爱与角色扮演比普通日程模拟更重视信息差和主观立场，因此系统进一步把记忆限定为当前角色私有，并区分亲眼观察、角色推断和他人传闻。每条重

		要记忆还需要可见事件作为证据，避免模型把想象自动写成事实。 这部分最终演化为 Character Profile、Relationship Lens、角色私有记忆、公开对话与阶段历史共同组成的角色上下文。
角色 AI 如何能够最大程度上像真人一样控制自己的行动？ AI 如何不越权？	Google DeepMind 的 Concordia 采用了类似桌面角色扮演游戏的设计：Agent 描述自己想做什么，Game Master 负责解释环境、判断行动是否符合物理与社会条件，并将其转化为世界结果。相关论文为 Generative agent-based modeling with actions grounded in physical, social, or digital space using Concordia。	CP 跳动最重要的权限分层设计借鉴了论文： 代码块 <pre>1 Character Agent 提出意图 2 ↓ 3 Interaction Runtime 校验与执行 4 ↓ 5 Relationship Judge 结算关系影响 6 ↓ 7 Memory Runtime 提交角色记忆</pre> Character Agent 可以提出靠近、说话、拒绝或离开，但不能直接修改坐标、关系数值和另一角色的状态。模型输出被视为不可信提案，只有通过 Runtime 后才会成为正式世界事实。 CP 跳动没有直接使用一个全能 Game Master，因为一个同时控制环境、关系和角色结果的模型仍然可能越权。因此，Concordia 的 Game Master 被进一步拆成 Interaction Runtime、Relationship Judge、Memory Runtime 与 Story Runtime，让物理执行、关系结算、记忆提交和剧情事件拥有不同权限。
如果玩家长时间运行项目，如何维护角色上下文一致性和稳定性？	Agentopia 与论文 Agentopia: Long-Term Life Simulation and Learning in Agent Societies 研究的是大规模、长时间的 Agent 社会模拟。其代码将角色扮演 Agent 与世界模拟系统分开，并为每个角色维护 profile、对话记录、长期历史以及由角色管理的 memory scratchpad。Scratchpad 又按照 general、characters 和其他主题拆分。	CP 跳动实际借鉴并改造了： • Roleplay：保存长期稳定的人格与角色身份； • Stage：只描述当前回合、现场能力与行为边界； • Message History：保留有限的近期逐字互动； • 按角色和主题拆分私有记忆文件； • 记忆修改前先读取原内容，而不是让模型直接覆盖； • 长期运行数据使用追加式历史，而不是只保留最后结果。

在此基础上，CP 跳动增加了 `baseRevisionId`、证据事件、访问记录、置信度与显著度。Agent 只有真实读取了最新 revision，且引用了自己能够看到的事件，才允许提交新记忆。

没有直接照搬 Agentopia 的原因是：它面向百人、年度尺度的社会模拟，而 CP 跳动关注少量角色之间高密度、细粒度的情感互动。因此 CP 跳动保留更短的逐字对话、更严格的私人信息隔离，以及身体接触和双人行为的独立同意流程。